What is claimed is:

1. An apparatus comprising:

first storage logic to store a first portion of a dataword;

second storage logic to store a second portion of the dataword;

determination logic to determine a condition of the dataword;

third storage logic to store a result generated by the determination logic; and

selection logic to select, based on the contents of the third storage logic, one of

the contents of the second storage logic and a replacement value that

depends on the contents of a predetermined bit of the first storage logic.

2. The apparatus of claim 1, wherein the first portion of the dataword is the least

significant portion of the dataword.

3. The apparatus of claim 1, wherein the condition of the dataword is that the value of

each bit of the second portion of the dataword is the same as the value of the most

significant bit of the first portion of the dataword.

4. The apparatus of claim 1, wherein each bit of the replacement value is equal to the

contents of the predetermined bit of the first storage logic.

5. The apparatus of claim 1, wherein the predetermined bit of the first storage logic is

to store the most significant bit of the first portion of the dataword.

6. The apparatus of claim 1, wherein the first storage logic and the second storage

logic are included in one of a register, an issue queue, a data cache, and a data holding latch.

7. An apparatus comprising:

    first storage logic to store a first portion of a dataword;

    second storage logic to store one of the first portion of the dataword and a second portion of the dataword;

    determination logic to determine a condition of the dataword;

    third storage logic to store a result generated by the determination logic;

    first selection logic to select, based on the result generated by the determination logic, one of the first portion of the dataword and the second portion of the dataword to store in the second storage logic; and

    second selection logic to select, based on the contents of the third storage logic, one of the contents of the second storage logic and a replacement value.

8. The apparatus of claim 7, wherein the condition of the dataword is that the value of each bit of the second portion of the dataword is the same as the value of each other bit of the second portion of the dataword.

9. The apparatus of claim 7, wherein the condition of the data is that the value of the second portion of the dataword is zero.

10. The apparatus of claim 7, wherein the replacement value is zero.

11. The apparatus of claim 7, wherein the first storage logic and the second storage logic are included in one of a register, an issue queue, a data cache, and a data holding latch.

12. The apparatus of claim 7, further comprising error detection logic to compare the contents of the first storage logic and the second storage logic.

13. The apparatus of claim 12, wherein the error detection logic is to compare the contents of the first storage logic and the second storage logic responsive to the contents of the third storage logic.

14. The apparatus of claim 7, further comprising:

fourth storage logic to store one of the first portion of the dataword and a third portion of the dataword;

third selection logic to select, based on the result generated by the determination logic, one of the first portion of the dataword and the third portion of the dataword to store in the fourth storage logic;

a datapath for reading a stored value from the first storage logic; and

error correction logic to compare the contents of the first storage logic, the second storage logic, and the fourth storage logic and, if the contents of the second storage logic are different from the contents of the first storage logic and the same as the contents of the fourth storage logic, to provide to the datapath the contents of the second storage logic instead of the contents of the first storage logic.

15. A method comprising:

    determining that a dataword to be stored is a narrow value;

    storing the required portion of the narrow value;

    storing an indication that the stored data represents a narrow value;

    reading the required portion of the narrow value from storage; and

    providing a replacement value for the remainder of the narrow value.

16. The method of claim 15, further comprising redundantly storing the required portion of the narrow value.

17. The method of claim 16, further comprising:

    reading the redundantly stored required portion of the narrow value from storage; and

    comparing the required portion of the narrow value to the redundantly stored portion of the narrow value to check for errors.

18. The method of claim 17, further comprising correcting an error in the required portion of the narrow value by providing the redundantly stored required portion of the narrow value instead of the required portion of the narrow value from storage.

19. A method comprising:

    determining that a dataword to be stored includes an irreplaceable portion and a replaceable portion;

    storing the irreplaceable portion;

    storing a redundant copy of the irreplaceable portion;

storing an indication that the stored data represents a dataword including a

replaceable portion;

reading the irreplaceable portion from storage;

reading the redundant copy of the irreplaceable portion from storage; and

comparing the irreplaceable portion from storage and the redundant copy of

the irreplaceable portion from storage to check for errors.

20. The method of claim 19, where the replaceable portion has a value of zero.

21. A system comprising:

a memory; and

a processor including:

first storage logic to store a first portion of a dataword;

second storage logic to store a second portion of the dataword;

determination logic to determine a condition of the dataword;

third storage logic to store a result generated by the determination

logic; and

selection logic to select, based on the contents of the third storage

logic, one of the contents of the second storage logic and a

replacement value that depends on the contents of a predetermined

bit of the first storage logic.